A hierarchical clustering method for estimating copy number variation

BAIFANG XING
Genetics and Genome Biology, Hospital for Sick Children, Toronto, Ontario, Canada and Samuel Lunenfeld Research Institute of Mount Sinai Hospital, Toronto, Ontario, Canada

CELIA M. T. GREENWOOD∗
Genetics and Genome Biology, Hospital for Sick Children, Toronto, Ontario, Canada and Department of Public Health Sciences, University of Toronto, Toronto, Ontario, Canada
celia.greenwood@utoronto.ca

SHELLEY B. BULL
Samuel Lunenfeld Research Institute of Mount Sinai Hospital, Toronto, Ontario, Canada and Department of Public Health Sciences, University of Toronto, Toronto, Ontario, Canada

SUMMARY
Microarray technologies allow for simultaneous measurement of DNA copy number at thousands of positions in a genome. Gains and losses of DNA sequences reveal themselves through characteristic patterns of hybridization intensity. To identify change points along the chromosomes, we develop a marker clustering method which consists of 2 parts. First, a “circular clustering tree test statistic” attaches a statistic to each marker that measures the likelihood that it is a change point. Then construction of the marker statistics is followed by outlier detection approaches. The method provides a new way to build up a binary tree that can accurately capture change-point signals and is easy to perform. A simulation study shows good performance in change-point detection, and cancer cell line data are used to illustrate performance when regions of true copy number changes are known.

Keywords: Array CGH; Change-point; Genomic copy number; Outlier detection; Permutation.

1. INTRODUCTION
DNA copy number alterations in tumor cells are key genetic events in the development and progression of human cancers (Tirkkonen and others, 1998; Lengauer and others, 1998; Foronzan and others, 2000). These alterations are typically a result of genomic events producing gains and losses of chromosomes or...
chromosomal regions. Losses and gains of DNA can contribute to alterations in the expression of tumor suppressor genes and oncogenes, respectively. Therefore, the identification of DNA copy number alterations in tumor genomes may help to discover critical genes associated with cancers and, eventually, to improve therapeutic approaches. A variety of techniques have been developed to screen genomes for all possible regions with DNA copy number alterations. For example, comparative genomic hybridization (CGH) (Kallioniemi and others, 1992, 1994), high-resolution array-based CGH (Pinkel and others, 1998; Snijders and others, 2001), representational difference analysis (Lisitsyn and others, 1993), and single nucleotide polymorphism (SNP) arrays (Zhao and others, 2004) can all be used to assess copy number.

The goal in analyzing DNA copy number data is to partition the whole genome into regions with the same underlying copy number and, subsequently, to quantify the copy number in each region. Therefore, locating copy number changes is an important step in the analysis of DNA copy number data. Various statistical methods have been proposed to address this problem. The software CGH-Plotter uses moving median and mean filtering, $k$-means clustering, and a dynamic programming algorithm (Autio and others, 2003). Jong and others (2003, 2004) proposed a genetic local search algorithm to maximize a likelihood with a penalty term containing the number of change points. Other proposals for change-point detection methods include a dynamic programming algorithm with a penalized likelihood (Picard and others, 2005) and a weighted likelihood function using adaptive weight smoothing (Hupe and others, 2004). Olshen and others (2004) proposed a modified binary segmentation procedure, called circular binary segmentation (CBS), to look for 2 breakpoints at a time in the data arranged in a circle. Wang and others (2005) proposed a simple but effective method based on hierarchical clustering along the chromosomes (CLAC) to select regions of interest while simultaneously controlling the false discovery rate (FDR). Fridlyand and others (2004) used hidden Markov models (HMMs) to estimate the underlying copy numbers (the hidden states). In Hsu and others (2005), a smoothing algorithm was given that denoises the data using a wavelets approach.

A recent paper (Lai and others, 2005) described an extensive study comparing several of these methods and pointed out that some segmentation methods, especially the CGH segmentation method (Picard and others, 2005) and CBS (Olshen and others, 2004), appeared to perform consistently well whether the noise level was low or high. Willenbrock and Fridlyand (2005) also did a comparison study of HMMs, CBS, and the Hupe method and concluded that CBS has the best operational characteristics in terms of its sensitivity and FDR. In Picard’s method, however, the CGH profile is assumed to follow a Gaussian signal, an assumption that may be violated, while CBS is not designed to detect single outliers. For processing a large number of high-resolution arrays, the speed of many algorithms becomes an issue. Both HMMs and CBS can be slow and difficult to apply to very high–density oligonucleotide arrays.

In this report, we seek approaches to segment the chromosomes into regions of equal copy number that can include either a single marker or multiple contiguous markers. We propose a method, which we call the “circular clustering tree test statistic” (CCTTS), and following the construction of the statistics, we apply outlier detection approaches to identify change points along the chromosomes. Our method is novel in that it provides a new way to build up a binary tree that can accurately capture change-point signals and is easy to perform.

The rest of the paper is organized as follows. The CCTTS algorithm and properties are given in Section 2. In Section 3, we provide several change-point detection approaches based on the scores produced by the CCTTS algorithm. In Section 4, the proposed methods are evaluated by simulation studies. The method is applied to data from cell lines with known copy number changes in Section 5. We summarize our conclusions and discuss open questions in Section 6.

2. Methods

It is clear that estimating the locations of regions with aberrant DNA copy numbers is closely connected to the problem of finding the locations of change points (Olshen and others, 2004). This makes
change-point methods a natural framework to approach the analysis of array DNA copy number data. Let \( Y_1, Y_2, \ldots \) be a sequence of random variables that correspond to markers positioned sequentially along the genome. In general, an index \( p: i \leq p < j \) is called a change point if \( Y_i, \ldots, Y_p \) have a common distribution and \( Y_{p+1}, \ldots, Y_j \) have a different common distribution. Intuitively, a point can be identified as a change point when the difference in signal intensity is large between 2 adjacent segments demarcated by the point. The correct choice for the sizes of the 2 adjacent segments is an important issue when the signal difference is measured. Segments that are either too small or too large will reduce the sensitivity of change-point detection methods. Our proposed method—the CCTTS—automatically gives 2 “optimal” adjacent segments (or clusters) with which to measure the difference at any point along a chromosome. The idea is to build a hierarchical cluster-style tree from the bottom upward such that the gain/normal/loss regions are separated into different branches. In this procedure, the tree building and the evaluation of statistics at any point are carried out simultaneously. Furthermore, the statistics associated with the tree fully reflect the information involved in the tree structure, including the magnitude of the difference between any 2 adjacent clusters and the order of the merging procedure. These statistics can then be used for detecting the most likely change points along a chromosome.

2.1 The CCTTS algorithm

The CCTTS algorithm uses a bottom-up strategy that generates a binary tree to represent the similarities in the observed copy number intensities along a chromosome. The algorithm begins with the data arranged in a circle, obtained by connecting the 2 ends of the chromosome together, with every observation representing a singleton cluster. At each of the \( N-1 \) steps, the closest 2 adjacent clusters are merged into a single cluster, where closest is defined by the smallest difference in copy number intensities, and the differences at the new cluster boundaries are updated. Hence, after each step there is one less cluster.

Let \( Y_1, Y_2, \ldots, Y_N \) be a sequence of random variables for the log2 relative copy number intensities at \( N \) marker positions. We define a signed distance to measure the difference between any 2 adjacent clusters as

\[
D(\{Y_{1i}\}_{i=1}^{n}, \{Y_{2i}\}_{i=1}^{m}) = \frac{\bar{Y}^{(1)} - \bar{Y}^{(2)}}{\sqrt{1/n + 1/m}},
\]

where markers \( Y_{11}, \ldots, Y_{n1} \) are adjacent to \( Y_{12}, \ldots, Y_{m2} \). Note that \( D(\cdot, \cdot) \) is simply a normal test statistic for the data with variance 1. Here, the implicit assumption that the data have a common variance (without loss of generality, 1) can be relaxed for segmentation methods. We call index \( j \) (\( k \)) the left (right) boundary of a cluster \( \{Y_{j+1}, \ldots, Y_k\} \) (when \( j = 0 \), in a circular arrangement, the left boundary of a cluster \( \{Y_1, \ldots, Y_k\} \) is defined to be \( N \)). The CCTTS iterative algorithm is given in detail in the following and is illustrated in Figure 1.

1) Arrange the data in a circle so that the first point is connected to the last. That means \( Y_{N+1} = Y_1 \) and \( Y_0 = Y_N \). Define \( N \) clusters each consisting of one point. Let \( I = \{1, 2, \ldots, N\} \) be the set of indices corresponding to each of the \( N \) clusters, and define a set of statistics that will be used to decide if point \( i \) is a change point. To start, use

\[
S_i = D(\{Y_i\}, \{Y_{i+1}\}), \quad i = 1, \ldots, N,
\]

which is the distance between each pair of adjacent observations.
2) Let $l = \arg\min_{i \in I} |S_i|$. Then merge the 2 adjacent clusters $\{Y_i\}$ and $\{Y_{i+1}\}$, remove index $l$ from $I$, and update $S_{i-1}$ and $S_{i+1}$ in the following way:

$$
S_{i-1} \leftarrow \begin{cases} 
S_{i-1}, & |S_{i-1}| > |D(\{Y_{i-1}\}, \{Y_i, Y_{i+1}\})|, \\
D(\{Y_{i-1}\}, \{Y_i, Y_{i+1}\}), & |S_{i-1}| \leq |D(\{Y_{i-1}\}, \{Y_i, Y_{i+1}\})|.
\end{cases}
$$

$$
S_{i+1} \leftarrow \begin{cases} 
S_{i+1}, & |S_{i+1}| > |D(\{Y_i, Y_{i+1}\}, \{Y_{i+1}\})|, \\
D(\{Y_i, Y_{i+1}\}, \{Y_{i+2}\}), & |S_{i+1}| \leq |D(\{Y_i, Y_{i+1}\}, \{Y_{i+2}\})|.
\end{cases}
$$

3) In general, merge the 2 adjacent clusters $C_1 = \{Y_{j+1}, \ldots, Y_q\}$ and $C_2 = \{Y_{q+1}, \ldots, Y_k\}$ with the smallest $|S_i|$: $|S_q| = \min_{i \in I} |S_i|$. Then remove index $q$ from $I$. If the data still contain more than one cluster, update $S_j$ and $S_k$ in the following way:

- Let LC and RC be the left and right adjacent clusters to the cluster $C = \{Y_{j+1}, \ldots, Y_k\}$ we just formed, respectively (see Figure 1).
- Update $S_j$ using

$$
S_j \leftarrow \begin{cases} 
S_j, & |S_j| > |D(LC, C)|, \\
D(LC, C), & |S_j| \leq |D(LC, C)|.
\end{cases}
$$

(2.2)

- Update $S_k$ using

$$
S_k \leftarrow \begin{cases} 
S_k, & |S_k| > |D(C, RC)|, \\
D(C, RC), & |S_k| \leq |D(C, RC)|.
\end{cases}
$$

(2.3)
4) Repeat Step (3) until one big cluster is obtained. After merging the last 2 clusters, updating \( S \) is not relevant and is not performed. At that point there will be \( N \) statistics \( S_j, j = 1, \ldots, N \).

In Step (2) or (3), if there are ties for the minimum value \( \min_{i \in I} |S_i| \), say, \( |S_{q_1}| = \cdots = |S_{q_m}| = \min_{i \in I} |S_i| \), then indices \( q_1, \ldots, q_m \) are simultaneously removed from \( I \), all corresponding pairs of adjacent clusters are merged, and all statistics \( S_i \) on the boundaries of these newly formed clusters are updated as in Step (2) or (3). Note that 2 boundary points may correspond to the merging of more than 2 adjacent clusters.

The simple example in Table 1 is used to illustrate the CCTTS algorithm. The first row shows a data set with 5 points. The algorithm starts by defining 5 clusters: \( \{Y_1\}, \{Y_2\}, \{Y_3\}, \{Y_4\}, \{Y_5\} \) and \( I = \{1, 2, 3, 4, 5\} \). \( S_i = D([Y_i], [Y_{i+1}]) \) \( (i = 1, 2, 3, 4, 5) \) are calculated and given in the second row after connecting the data in a circle. In the first step, \( \arg\min_{i \in I}|S_i| = 5 \) is removed from \( I \), and \( \{Y_5\} \) and \( \{Y_1\} \) are merged together to obtain 4 clusters: \( \{Y_2\}, \{Y_3\}, \{Y_4\}, \{Y_5, Y_1\} \). To update \( S_4 \) and \( S_1 \) corresponding to the boundaries of the newly formed cluster, \( D([Y_4], [Y_5, Y_1]) \) is compared to \( S_4 = -0.495 \) and \( D([Y_5, Y_1], \{Y_2\}) \) is compared to \( S_1 = -0.566 \). Both \( S_1 \) and \( S_4 \) are updated to new values that are larger in absolute values as seen in the third row. The remaining values of \( S_i, i \in I = \{1, 2, 3, 4\} \), are unchanged. Then, this procedure of computing \( \arg\min_{i \in I}|S_i| \), updating \( I \), merging 2 adjacent clusters together, calculating \( D(\cdot, \cdot) \), and updating \( S_i \) is repeated until one cluster is formed. The final merging order \( 5, 2, 4, 1, 3 \) matches the magnitude of the final statistics \( S_i \) (in absolute value).

In addition to treating the data as a circle, there is a substantial difference between the CCTTS algorithm and the conventional clustering approach (Wang and others, 2005) with respect to how \( S_j \) and \( S_k \) are updated in Step (2) or (3), for a given difference measure. In the conventional approach, \( S_j \) and \( S_k \) are simply updated by \( D(LC, C) \) and \( D(C, RC) \), respectively, which ignores the previous statistic \( S^O \) at a marker when updating the statistic to \( S^N \), after merging clusters \( C_1 \) and \( C_2 \). This leads to a suboptimal choice of statistics. To clarify, let \( S^O_j \) be the difference measurement between clusters \( LC \) and \( C_1 \), and let \( S^N_j \) be the difference between \( LC \) and \( C \). Suppose \( Y_j \) is a true change point and \( |S^O_j| > |S^N_j| \), then our algorithm assigns the larger statistic \( S^O_j \) to marker \( j \), whereas the standard algorithm would assign the smaller statistic \( S^N_j \). This also means that there can be inconsistency between the order in which indices are removed from \( I \) and the magnitude of the resulting statistics \( |S_i| \). With the usual tree-building procedure, \( |S_j| \) and \( |S_k| \) may be smaller than \( |S_q| \), even though position \( q \) is merged before positions \( j \) and \( k \). The merging order indicates that \( Y_q \) is less likely to be a change point than \( Y_j \) and \( Y_k \). Our method of updating difference measurements \( S_j \) in the CCTTS algorithm will produce a sequence of statistics \( S_1, S_2, \ldots, S_N \), ordered by their position in the clustering process.

The CCTTS algorithm provides a way to calculate the statistics \( S_i \) and also leads to 2 “optimally sized” adjacent clusters to achieve a statistic value (see Figure 2). Updating procedures (2.2) and (2.3)

| \( Y_i \) | 0.8 | 1.6 | 1.3 | 0.2 | 0.9 | \( I \) | \( \arg\min_{i \in I}|S_i| \) |
|---------|-----|-----|-----|-----|-----|-------|----------------|
| \( S_j \) | -0.566 | 0.212 | 0.778 | -0.495 | 0.071 | \{1, 2, 3, 4, 5\} | 5 |
| \( D(\cdot, \cdot) \) | -0.612 | -0.531 | 0.212 | 0.778 | -0.531 | \{1, 2, 3, 4\} | 2 |
| Update \( S_i \) | -0.612 | 0.212 | 0.778 | -0.531 | | \{1, 2, 3, 4\} | 2 |
| \( D(\cdot, \cdot) \) | -0.6 | 1.021 | | | | | |
| Update \( S_i \) | -0.612 | 1.021 | 0.778 | -0.531 | | | |
| \( D(\cdot, \cdot) \) | -0.895 | 0.895 | 0.212 | 1.021 | -0.531 | | |
| Update \( S_i \) | -0.895 | 0.212 | 1.021 | -0.531 | 0.071 | | |
A hierarchical clustering method for estimating copy number variation

2.2 Properties

Under the null hypothesis of no change points, the final statistics \( S_1, S_2, \ldots, S_N \) have the following properties:

a) If \( Y_1, \ldots, Y_N \) are iid, then all \( S_i \) \( (i = 1, \ldots, N) \) are identically distributed, but not independent.

b) If \( \{Y_1, \ldots, Y_N\} \) come from a normal population, then all \( S_i \) \( (i = 1, \ldots, N) \) are approximately identically normally distributed, but not independent.

The clustering tree built by the CCTTS algorithm is based on a circular arrangement of the data. Property (a) is a natural consequence of this circular symmetric arrangement. In Appendix A, we give a simple example of a sequence data set containing only 3 points \( Y_1, Y_2, \) and \( Y_3 \) and illustrate the functional relationships between the statistic scores \( S_i \) and the data set \( \{Y_1, Y_2, Y_3\} \). In general, for a given data set...
arranged in a sequence: \( Y_1, Y_2, \ldots, Y_N \), the method for calculating \( S_i \) is a function of \( Y_1, Y_2, \ldots, Y_N \). So we write \( S_1 = f(Y_1, Y_2, \ldots, Y_N) \), where the subscript 1 on \( S \) corresponds to the first location, \( Y_1 \). On the other hand, for any \( j : 1 < j \leq N \), if the data are rearranged as \( Y_j, Y_{j+1}, \ldots, Y_N, Y_1, \ldots, Y_{j-1} \), then, according to the previous notation, the CCTTS algorithm gives statistic \( S'_j \) corresponding to the first location of \( Y_j \): 

\[
S'_j = f(Y_j, Y_{j+1}, \ldots, Y_N, Y_1, \ldots, Y_{j-1}).
\]

Since both data sets \( Y_1, Y_2, \ldots, Y_N \) and \( Y_j, Y_{j+1}, \ldots, Y_N, Y_1, \ldots, Y_{j-1} \), when arranged in a circle, are exactly the same, \( S_j = S'_j = f(Y_j, Y_{j+1}, \ldots, Y_N, Y_1, \ldots, Y_{j-1}) \). Therefore, \( S_1 = f(Y_1, Y_2, \ldots, Y_N) \) and \( S_j = f(Y_j, Y_{j+1}, \ldots, Y_N, Y_1, \ldots, Y_{j-1}) \) \((j = 2, 3, \ldots, N)\) are identically distributed because \( Y_1, \ldots, Y_N \) are iid. Property (a) plays a significant role when analyzing non-normal data for change-point detection, since the symmetry can be exploited to reduce the number of permutations needed to produce a reference distribution.

Property (b) has been established empirically by examining normal \( Q-Q \) plots of simulated data. From property (a), we only need to illustrate that \( S_1 \) is approximately normally distributed. Let \( \{Y_{i1}, \ldots, Y_{iN}\}, i = 1, \ldots, n \), be \( n \) data sets generated iid from \( N(0, 1) \). The statistics \( \{S_{i1}, \ldots, S_{iN}\} \) were computed from the \( i \)th data set by the CCTTS algorithm. Then \( \{S_{11}, S_{21}, \ldots, S_{n1}\} \) is a sample drawn from the distribution of \( S_1 \). Figure 3 shows normal \( Q-Q \) plots for \( n = 10000 \) replications with number of

![Q-Q plots](http://biostatistics.oxfordjournals.org/)

Fig. 3. Normal \( Q-Q \) plots with respect to \( S_1 \) from 10000 simulated data sets. The dash-dotted line is a robust linear fit of the sample order statistics which passes through the first and third quartiles. The slope of the solid line is 1 and represents the standard normal distribution. Simulations are applied for different data size \( N \). The plots demonstrate approximate normality of the statistics but with larger variances than standard normal.
markers \( N = 50, 100, 200, 400, 800, 1600 \) and indicates that \( S_1 \) is approximately normally distributed. In our simulations, we simultaneously calculated sample correlation coefficients between \( S_1 \) and \( S_{1+j} \) (\( j > 0 \)) which, because of the symmetric circular arrangement and property (a), also estimate the correlation coefficients for any 2 statistics \( S_i \) and \( S_{i+j} \). When \( j = 1 \), for different \( N \), all these correlation coefficients were between \(-0.38 \) and \(-0.35 \). When \( j > 1 \), correlations between \( S_i \) and \( S_{i+j} \) are small (e.g. correlations between \( S_i \) and \( S_{i+2} \) are around \(-0.066 \)), and these statistics could be treated as approximately independent. Due to the negative correlations between \( S_i \) and \( S_{i+1} \), a histogram of all statistics \( S_1, \ldots, S_N \) jointly has larger tails than would be expected from the marginal approximate normality of any one statistic \( S_i \). Property (b) makes it possible to analyze normal data for change-point detection easily and efficiently.

3. Change-point detection

We assume that the observed \( Y_i \) is a realization of the true log \( \log_2 \) relative copy number \( \alpha + \beta_i \) at marker \( i \) plus random noise,

\[
Y_i = \alpha + \beta_i + \epsilon_i, \quad i = 1, \ldots, N,
\]

where \( N \) is the number of markers on a given chromosome and \( \epsilon_i \) are iid with mean 0. Under the null hypothesis of no change points \( H_0: \beta_i = 0, i = 1, \ldots, N \), property (a) implies that \( S_1, \ldots, S_N \) have the same distribution. As discussed in Section 2.2, the problem of change-point identification can be reduced to the problem of finding extremely large \( |S_i| \) that deviate from the pattern of \( \{S_1, \ldots, S_N\} \). In other words, our task becomes the identification of the outliers among the scores \( \{S_1, \ldots, S_N\} \). Interpretation of identified change points must take into account the artificial circular data. For example, if the number of detected change points is one, we must conclude that the original data contain no change points, since circular data with only one change point are still one cluster. For simplicity, we will not exclude a detected change point that is the last point on a chromosome.

We discuss and introduce several different approaches for outlier detection, separately for normal and non-normal data.

3.1 Normally distributed data

We assume that the data \( \{Y_1, \ldots, Y_N\} \) are normally distributed with a common variance. The scores \( S_1, \ldots, S_N \) are computed according to the CCTTS. Under the null hypothesis of no change points, \( S_1, \ldots, S_N \) are each approximately normally distributed (see property (b)). Therefore, some approaches used to identify outliers for normal data are appropriate here, even though \( S_1, \ldots, S_N \) are negatively correlated between 2 adjacent \( S_i \). To perform this procedure, we apply the simple \( d \) standard deviation method, where the value of \( d \) can be user defined.

**Approach 1: outlier detection method (ODM).** \( S_1, \ldots, S_N \) are sorted in decreasing absolute values: \( |S_{p_1}| \geq |S_{p_2}| \geq \cdots \geq |S_{p_N}| \). The method starts with the assumption of no outliers in \( S_1, \ldots, S_N \), then examines and determines the first most likely outliers. If outliers are detected, then they are removed from the set of scores \( S = \{S_1, \ldots, S_N\} \). The procedure will be repeated for the remaining scores until no more outliers can be found.

1) Set \( j = 0 \). Calculate the mean \( m \) and standard deviation \( s \) of all the scores.
2) If \( |S_{p_1} - m| \leq d \cdot s \), stop. Otherwise, identify the largest \( l \) such that \( |S_{p_1} - m|, \ldots, |S_{p_l} - m| > d \cdot s \). Then remove \( S_{p_1}, \ldots, S_{p_l} \) from \( S \), recalculate the mean \( m \) and standard deviation \( s \) for the remaining scores in \( S \), and then update \( j \leftarrow j + l \).
In general, if \[ |S_{p_{j+1}} - m| \leq d \cdot s, \] stop. Otherwise, identify the largest \( l \) such that \[ |S_{p_{j+1}} - m|, \ldots, |S_{p_{j+l}} - m| > d \cdot s. \]

4) Remove \( S_{p_{j+1}}, \ldots, S_{p_{j+l}} \) from \( S \), then recalculate the mean \( m \) and standard deviation \( s \) for the remaining scores in \( S \).

5) \( j \leftarrow j + l \), go to Step (3).

After completing this procedure, if \( j \leq 1 \), there are no change points; otherwise, we find change points \( Y_{p_1}, \ldots, Y_{p_j} \).

The value of \( d \) can be user defined based on some domain or data knowledge, or theoretically using Chebyshev’s theorem (Barnett and Lewis, 1994). In practice, \( d \) is usually chosen between 3 and 6 (Maletic and Marcus, 2000). From our experience, values between 3.5 and 4 were found to generate the best results with fewer false positives and false negatives in general.

Under the null hypothesis of no change points, the negative correlations between \( S_i \) and \( S_{i+1} \) and approximate independence between \( S_i \) and \( S_{i+j} \) (\( j > 1 \)) mean that a sample variance \( \hat{\sigma}^2 \) from \( S_1, \ldots, S_N \) overestimates the variance \( \sigma^2 \) of the marginal distribution for any one statistic \( S_k \). Therefore, outlier detection in the outlier detection method (ODM) is conservative. Outlier detection in \( S_1, \ldots, S_N \) with one or several (outliers) points removed is also expected to be conservative. However, this conservativeness is not a concern in most cases due to the following 2 facts: (i) a sample variance \( \hat{\sigma}^2 \) from \( S_1, \ldots, S_N \) is an asymptotically unbiased estimator of the marginal variance \( \sigma^2 \) with respect to the number of markers \( N \) (see Appendix B) and (ii) a user-defined value of \( d \) can be adjusted for outlier detection. Therefore, the negative correlations do not have much effect on ODM performance for large data sets. When the number of markers is small, we may choose a smaller value of \( d \) to adjust for the conservativeness.

### 3.2 Non-normal data

In the outlier detection method, we assumed that the data followed a normal distribution. For non-normal data, we will introduce 2 methods to identify change points based on permutation-derived reference distributions. The sequential detection method (SDM) is the most straightforward method to apply.

**Approach 2: Sequential detection method (SDM).** Assume the null hypothesis \( H_0 \) is true, that is, there are no change points. Using property (a), each statistic value \( S_i \) computed from position permuted data has the same distribution and, therefore, can be used as a contribution to the permutation distribution. For example, by doing 100 permutations on a data set of \( N = 200 \) markers, 20,000 null test statistics are obtained. This drastically reduces the number of permutations needed.

1) For the original data \( Y_1, \ldots, Y_N \), we calculate statistic scores according to CCTTS and sort them in decreasing absolute values: \( |S_{p_1}| \geq |S_{p_2}| \geq \cdots \geq |S_{p_N}| \).

2) Randomly permute the order of the markers \( M \) times (\( M \times N > 10,000 \)). For each permutation data set \( P \), collect all absolute statistic values \( |S_{p_1}^P|, \ldots, |S_{p_N}^P| \) and construct a reference distribution. Outliers are chosen if \( |S_{p_j}| \) is greater than the upper \( \alpha \)th quantile of the permutation distribution \( U_{\alpha} \). Let \( k = \max_j (|S_{p_j}| > U_{\alpha}) \).

3) If \( k \leq 1 \), there are no change points. Otherwise, the data can be partitioned into \( k \) or \( k + 1 \) segments separated by \( k \) change points \( Y_{p_1}, \ldots, Y_{p_k} \). The exact number of segments (\( k \) or \( k + 1 \)) depends on whether \( Y_N \) is chosen as a change point.

4) Steps (1) to (3) are recursively performed on each small segment until no more change points are identified.
Single-point regions of gain or loss may be identified by the occurrence of 2 adjacent change points. These may occur either because there was a very small region of copy number change on the chromosome that is measured by only one marker or because there was an error in the data at one marker. Suppose a chromosome with \( N \) markers has only one single-point region of gain/loss. Any permuted data from this chromosome also has one single-point region of gain/loss. That means 2 out of the \( N \) statistic values \( S_i \) in each permutation will be relatively large, and therefore the magnitude of quantile \( U_\alpha \) depends critically on \( N \). It will be difficult to detect change points for small values of \( N \). Therefore, since the SDM does not work well in this situation, we propose a modification to improve detection of single-point regions. Let \( s \) be the sample standard deviation of the original data. Suppose during Step (3) on a segment with mean \( m_s \) no change points were detected (\( k \leq 1 \)). We propose to examine the indices of \( S_{p_1} \) and \( S_{p_2} \). If they are adjacent in the circle, then point \( Y_{p_1} \) (or \( Y_{p_2} \), depending on the relative positions of \( p_1 \) and \( p_2 \)) may be a single-point region. If \( |Y_{p_1} - m_s| \leq 4s \), then we conclude that there are no change points on the segment; otherwise, we conclude that \( Y_{p_1} \) and \( Y_{p_2} \) are 2 change points.

**Approach 3: backward residual method (BRM).** The SDM works well and is robust to unequal variances. However, when segments become small, the SDM permutation approach can miss some change points. For example, look at a very simple data set: \( \{Y_1, \ldots, Y_9\} = \{0, 1, 1, 1, 0, 0\} \). It is obvious that there are 2 change points and the largest statistic, \( |S_{p_1}| \), is 1.2247. The probability of achieving this largest value in the permutation distribution is \( \frac{31 \times 1 \times 1}{2! \times 6!} = 0.1 \). No change points can be identified at the level \( \alpha < 0.1 \). Therefore, as an alternative approach, we developed a backward residual method (BRM) that uses the whole data set all the time rather than working with smaller segments. The algorithm starts in the same way as the SDM, but after some of the change points are detected, new permutations will be performed on the residual data instead of on the smaller segments.

1) As before, for the original data \( Y_1, \ldots, Y_N \), we calculate statistic scores according to CCTTS and sort them in decreasing absolute values: \( |S_{p_1}| \geq |S_{p_2}| \geq \cdots \geq |S_{p_N}| \).

2) Let \( k_0 = 1 \) (one segment) and \( \{X_1, \ldots, X_N\} = \{Y_1, \ldots, Y_N\} \).

3) Randomly permute the order of the data \( \{X_1, \ldots, X_N\} \) \( M \) times (\( M \times N > 10000 \)). For each permutation data set, collect all absolute statistic values to form a reference distribution. The threshold values can be chosen to be the upper \( \alpha \)th quantile of the permutation distribution, say \( U_\alpha \). Let \( k = \text{argmax}_j (|S_{p_j}| > U_\alpha) \).

4) If \( k \leq k_0 \), stop. Otherwise, \( k_0 \leftarrow k \) (\( k_0 \) detected change points or segments); denote \( m_i \) as the mean of the segment which contains observation \( Y_i \) and update \( \{X_1, \ldots, X_N\} \) to be the mean adjusted residual of \( \{Y_1, \ldots, Y_N\} \), so that \( X_i = Y_i - m_i, i = 1, \ldots, N \). Then go to Step (3).

Similar to the SDM, a modification can be made to deal with single-point regions in data sets. Let \( s \) be the sample standard deviation of the original data. In Step (4), when \( k \leq k_0 \), let \( Y_{p_l} \) and \( Y_{p_{l+1}} \) (\( l = 1 \) when \( k_0 = 1, l = k + 1 \) when \( k_0 > 1 \)) be the next 2 most likely change points. Identify if there are potential single-point regions produced by combining \( Y_{p_l} \) with previously detected change points or by examining \( Y_{p_l} \) and \( Y_{p_{l+1}} \). Suppose this potential single-point region \( \{Y_j\} \) lies in segment \( L \). We need to decide if this is a real single-point region. If \( |Y_j - m_L| \leq 4s \) (\( m_L \) is the mean of segment \( L \)), we conclude that there are no more change points. Otherwise, \( Y_{p_l} \) or both \( Y_{p_l} \) and \( Y_{p_{l+1}} \) are identified to be new change points.

Software written in MATLAB is available from the first author.

4. Simulations

In this section, we investigate through Monte Carlo simulations the performance of the proposed methods. Data containing regions of gain, normal, and loss copy numbers were generated from the model...
\[ y_i = \mu_i + \sigma \epsilon_i, \quad 1 \leq i \leq N, \] where \( N \) is the sample size, \( \mu \) is the mean, and \( \epsilon \) is distributed as \( N(0, 1) \). We assume one region of loss and one region of gain per chromosome. Within a loss region, all markers have the same state, with copy number reduced by 1. However, within the gain region we allow 3 possible states amplified by 1, 2, or 3 copies.

Following Pollack and others (2002), we set \( \mu \)'s for the copy numbers deleted by 1 and amplified by 1, 2, and 3 to be 
\[ d_1 = \log_2(0.75), \quad a_1 = \log_2(1.25), \quad a_2 = \log_2(1.5), \quad \text{and} \quad a_3 = \log_2(1.75), \] respectively, and we use \( \mu = 0 \) for normal regions. To mimic typical data, the following quantities were generated randomly: the position of loss/gain regions, the size of loss/gain regions, the number and allocation of the 3 amplified states, and the sizes of the 3 amplified regions within the regions of gain. In order to compare our methods and the CBS method, the simulated data sets do not contain single points of loss/gain. The details of the simulation are given in the following.

1) A data set is defined to contain a normal region, a region containing a loss, and a region containing a gain in random order.

2) To generate the 3 regions, each of size \( m \): \( Y_1, \ldots, Y_m \), the following parameters are used:
   a) Data from normal region: \( Y_1, \ldots, Y_m \sim N(0, \sigma) \).
   b) Data from region of loss:
      i) Initially generate \( Y_1, \ldots, Y_m \sim N(0, \sigma) \).
      ii) Generate the width \( w \) of a loss region with minimum width \( w = 2 \):
          \[ w \sim \text{Uni}(\{2, 3, \ldots, m\}) \].
      iii) Locate the loss interval in the region: \( Y_{i+1}, \ldots, Y_{i+w} \) by generating \( i \) from a uniform distribution:
          \[ i \sim \text{Uni}(\{0, 1, \ldots, m-w\}) \].
      iv) Let \( d_1 = \log_2(0.75) \) and \( Y_{i+j} \leftarrow Y_{i+j} + d_1, \quad j = 1, \ldots, w \).
   c) Data from region of gain:
      i) Initially generate \( Y_1, \ldots, Y_m \sim N(0, \sigma) \).
      ii) Generate the number of amplified states: \( n \sim \text{Uni}(\{1, 2, 3\}) \) and then randomly select the magnitude of the gains by sampling without replacement \( n \) times from \( \{1, 2, 3\} \) within the region of gain.
      iii) Generate the width of a region with minimum width 2 for 1 amplified state, 4 for 2 amplified states, and 6 for 3 amplified states:
          \[ w \sim \begin{cases} \text{Uni}(\{2, 3, \ldots, m\}), & \text{if } n = 1, \\ \text{Uni}(\{4, 5, \ldots, m\}), & \text{if } n = 2, \\ \text{Uni}(\{6, 7, \ldots, m\}), & \text{if } n = 3. \end{cases} \]
      iv) Locate the gain interval in the region: \( Y_{i+1}, \ldots, Y_{i+w} \) by generating \( i \) from a uniform distribution:
          \[ i \sim \text{Uni}(\{0, 1, \ldots, m-w\}) \].
      v) Divide the gain region into \( n \) segments, with the length of each segment at least 2 and chosen from a uniform distribution, and then randomly assign the \( n \) amplified states (selected by (ii)) to the \( n \) segments.
Fig. 4. Examples of 4 simulated data sets with $\sigma = 0.05$. Each data set is produced by randomly concatenating the 3 parts: $\cdot$, $\times$, and $+$ which contain normal, loss, and gain regions, respectively. In the gain regions, (a) has 1 state amplified by 3 copies, (b) has 2 states amplified by 1 and 2 copies, (c) and (d) have all 3 states, amplified by either 1, 2, or 3 copies.

vi) Let $a_1 = \log_2(1.25)$, $a_2 = \log_2(1.5)$, and $a_3 = \log_2(1.75)$. If segment $Y_{k+1}, \ldots, Y_{k+v}$ corresponds to the $i$th state, then $Y_{k+j} \leftarrow Y_{k+j} + a_i, j = 1, \ldots, v$.

3) The whole simulated data set is produced by randomly concatenating the 3 regions generated by (2) and examples are given in Figure 4.

One hundred data sets with 180 markers were simulated for $m = 60$ and for each value of $\sigma \in \{0.05, 0.10, 0.15, 0.20, 0.25, 0.5, 1\}$. In order to compare the simulation results across different values of $\sigma$, $Y_1, \ldots, Y_m \sim N(0, 1)$ are generated and $\sigma Y_1, \ldots, \sigma Y_m$ are used for different $\sigma$. Examples with $\sigma = 0.05$ from simulation are shown in Figure 4. If regions of loss and gain are in the middle, a data set with only one amplified state for a gain region leads to 4 change points, whereas a data set with all 3 amplified states leads to 6 change points. The total number of change points in 100 data sets for a given $\sigma$ is 500. We use $d = 3.5$ standard deviations to perform the ODM. For both the SDM and the BRM, $\alpha = 0.001$ is used to define outliers.

The number of change points detected is summarized in Table 2. For comparison, the results of the CBS without pruning (Olshen and others, 2004) and the CLAC (Wang and others, 2005) are included. In order to compare the CPU time, we coded the algorithm for CBS in MATLAB instead of the existing R package. The CLAC method was included for comparison since it also involves hierarchical clustering of points. CLAC was implemented with the existing R package (available at http://www-stat.stanford.edu/~wp57/CGH-Miner/) using default parameters. Because reference samples from
Table 2. Results from applying ODM, SDM, BRM, CBS, and CLAC to the simulated data. ODM is performed with \( d = 3.5 \) standard deviation. \( \alpha = 0.001 \) is used to define outliers for both SDM and BRM. CBS is applied without pruning.

<table>
<thead>
<tr>
<th>Number of</th>
<th>Method</th>
<th>( \sigma = 0.05 )</th>
<th>( \sigma = 0.10 )</th>
<th>( \sigma = 0.15 )</th>
<th>( \sigma = 0.20 )</th>
<th>( \sigma = 0.25 )</th>
<th>( \sigma = 0.50 )</th>
<th>( \sigma = 1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>True change points</td>
<td>ODM</td>
<td>495</td>
<td>426</td>
<td>337</td>
<td>268</td>
<td>203</td>
<td>59</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>SDM</td>
<td>461</td>
<td>416</td>
<td>334</td>
<td>264</td>
<td>207</td>
<td>58</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>BRM</td>
<td>496</td>
<td>428</td>
<td>337</td>
<td>272</td>
<td>212</td>
<td>61</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>CBS</td>
<td>453</td>
<td>407</td>
<td>337</td>
<td>264</td>
<td>218</td>
<td>69</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>CLAC</td>
<td>13</td>
<td>13</td>
<td>15</td>
<td>26</td>
<td>29</td>
<td>27</td>
<td>26</td>
</tr>
<tr>
<td>Change points correctly detected</td>
<td>ODM</td>
<td>3</td>
<td>33</td>
<td>82</td>
<td>115</td>
<td>128</td>
<td>78</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>SDM</td>
<td>(3, 0, 0)</td>
<td>(29, 3, 1)</td>
<td>(63, 13, 6)</td>
<td>(76, 26, 13)</td>
<td>(80, 29, 19)</td>
<td>(36, 25, 17)</td>
<td>(7, 5, 9)</td>
</tr>
<tr>
<td></td>
<td>BRM</td>
<td>(3, 0, 0)</td>
<td>(26, 1, 1)</td>
<td>(60, 13, 5)</td>
<td>(76, 23, 13)</td>
<td>(81, 25, 20)</td>
<td>(40, 27, 19)</td>
<td>(8, 8, 10)</td>
</tr>
<tr>
<td></td>
<td>CBS</td>
<td>(3, 0, 0)</td>
<td>(30, 3, 1)</td>
<td>(65, 14, 6)</td>
<td>(79, 26, 13)</td>
<td>(86, 30, 21)</td>
<td>(38, 27, 19)</td>
<td>(10, 8, 10)</td>
</tr>
<tr>
<td></td>
<td>CLAC</td>
<td>231</td>
<td>214</td>
<td>209</td>
<td>190</td>
<td>185</td>
<td>128</td>
<td>102</td>
</tr>
<tr>
<td>Change points mislocated (in position by 1, 2, 3)</td>
<td>ODM</td>
<td>2</td>
<td>41</td>
<td>81</td>
<td>117</td>
<td>169</td>
<td>363</td>
<td>467</td>
</tr>
<tr>
<td></td>
<td>SDM</td>
<td>(44, 175, 12)</td>
<td>(46, 149, 19)</td>
<td>(69, 115, 25)</td>
<td>(72, 95, 23)</td>
<td>(77, 82, 26)</td>
<td>(57, 51, 20)</td>
<td>(51, 33, 18)</td>
</tr>
<tr>
<td></td>
<td>BRM</td>
<td>(44, 175, 12)</td>
<td>(46, 149, 19)</td>
<td>(69, 115, 25)</td>
<td>(72, 95, 23)</td>
<td>(77, 82, 26)</td>
<td>(57, 51, 20)</td>
<td>(51, 33, 18)</td>
</tr>
<tr>
<td>Missed change points</td>
<td>ODM</td>
<td>36</td>
<td>56</td>
<td>88</td>
<td>124</td>
<td>167</td>
<td>356</td>
<td>460</td>
</tr>
<tr>
<td></td>
<td>SDM</td>
<td>1</td>
<td>38</td>
<td>78</td>
<td>110</td>
<td>151</td>
<td>355</td>
<td>459</td>
</tr>
<tr>
<td></td>
<td>BRM</td>
<td>39</td>
<td>54</td>
<td>90</td>
<td>120</td>
<td>147</td>
<td>344</td>
<td>459</td>
</tr>
<tr>
<td></td>
<td>CLAC</td>
<td>256</td>
<td>273</td>
<td>276</td>
<td>284</td>
<td>286</td>
<td>345</td>
<td>372</td>
</tr>
<tr>
<td>Additional change points detected</td>
<td>ODM</td>
<td>5</td>
<td>9</td>
<td>11</td>
<td>19</td>
<td>30</td>
<td>45</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>SDM</td>
<td>8</td>
<td>21</td>
<td>28</td>
<td>41</td>
<td>48</td>
<td>71</td>
<td>42</td>
</tr>
<tr>
<td></td>
<td>BRM</td>
<td>10</td>
<td>13</td>
<td>18</td>
<td>28</td>
<td>36</td>
<td>61</td>
<td>42</td>
</tr>
<tr>
<td></td>
<td>CBS</td>
<td>7</td>
<td>13</td>
<td>22</td>
<td>19</td>
<td>29</td>
<td>53</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>CLAC</td>
<td>65</td>
<td>83</td>
<td>85</td>
<td>88</td>
<td>89</td>
<td>101</td>
<td>422</td>
</tr>
</tbody>
</table>
the normal arrays are needed in the CLAC method, we simulated 100 reference data sets using a normal distribution with zero mean and variance $\sigma$, set to be the same as in the data to be analyzed.

The simulation results in Table 2 show that the proposed methods are competitive with the CBS method without pruning. In contrast, most of the change points detected by the CLAC are not in the right locations, and many change points are missed. This may be caused by the mean smoothing step in CLAC that reduces noise in the simulation data at the expense of blurring the edges of the loss/gain regions. When $\sigma$ is small, our methods appear to perform better than CBS. When $\sigma$ is large (e.g. $\sigma = 0.5$), SDM finds a few more false positives and ODM has a few more false negatives than CBS. As $\sigma$ increases, the data sets become more noisy, and the numbers of both false negatives and false positives from all methods increase. However, once $\sigma$ reaches very large values, the data sets appear approximately equal to those without change points and the number of additional change points reflects the false-positive rate in the absence of any change points. This can be seen in the last 2 columns in Table 2, where the number of additional change points detected for $\sigma = 1$ is less than for $\sigma = 0.5$ (except for CLAC).

To further assess the true-positive and false-positive rates, we applied CCTTS-based methods in additional simulations with $\sigma$ gradually increasing toward infinity. We note that, from the viewpoint of change-point detection, analyzing a data set $y_1, y_2, \ldots, y_N$ is equivalent to analyzing $\lambda y_1, \lambda y_2, \ldots, \lambda y_N$ for any nonzero constant $\lambda$. Therefore, for any fixed $\sigma$, analyzing a simulation data set $y_i = \mu_i + \sigma \epsilon_i$, $1 \leq i \leq N$,
yields the same results as analyzing the data set \( y_i / \sigma = \mu_i / \sigma + \epsilon_i, \ 1 \leq i \leq N \). As \( \sigma \to \infty \), the data set becomes one without change points. In Figure 5, we plot change-point detection rates against \( \sigma \). We define true positives to be both change points correctly detected and those detected but mislocated. The results indicate that the number of change points correctly detected and the true-positive rates decrease as \( \sigma \) increases. However, among the true positives, the number of change points mislocated increases when \( \sigma \) is small and then decreases as \( \sigma \) becomes large. Similarly, for small values of \( \sigma \), the false positives increase due to the increased noise. When \( \sigma \) reaches a certain value (around 0.5), the number of false positives begins to decrease and eventually approaches a limiting value corresponding to \( \sigma = \infty \) (data without change points).

We also compared the CPU time consumed (with MATLAB software) to perform these analyses. For analyzing 100 data sets with \( \sigma = 0.15 \), the ODM, SDM, and BRM consumed about 0.02%, 6.5%, and 3.3% of the CPU time used by the CBS method, respectively.

5. Application to a Data Set of Fibroblast Cell Lines

To further assess the performance of our methods, we applied them to a set of CGH data with known DNA copy number changes (Snijders and others, 2001). The data consisted of single experiments on 15 fibroblast cell lines. The variable used for analysis was the normalized average of \( \log_2 \) of the test over reference ratio. For 6 of the cell lines, the copy number alteration encompassed the whole chromosome. Therefore, we first limited our analysis to the other 9 cell lines and tested one chromosome at a time for change points. The ODM was performed with both \( d = 3.5 \) and \( d = 4 \). For SDM and BRM, we used thresholds based on \( \alpha = 0.001 \) and 0.0005. Results can be found in Table 3. Outcomes from applying the CBS method reported by Olshen and others (2004) are included for comparison.

In Table 3, “yes” for ODM, SDM, and BRM means that the alteration was found for the particular cell line and chromosome at the given \( \alpha \) level, although in some cases, the locations of the change points may be 1 or 2 positions away from the real ones. Some additional change points may also be identified, which, as we will discuss further, can be eliminated through a “pruning” procedure. “Yes” for CBS is taken from Olshen and others (2004), but the localization relative to the truth was not given. “No” means that the alteration was not found. “Single” and “false” represent the number of chromosomes where change points were found that do not correspond to known alterations. The difference between “single” and “false” is that “single” includes only the chromosomes where all detected change points were single-point regions, whereas “false” includes regions that are larger than single points. Using the \( d = 4 \) criterion, ODM gives better results than other methods. Outcomes for both SDM and BRM show that there is a little difference between \( \alpha = 0.001 \) and 0.0005. Unlike the CBS application, our methods were directly applied to the original data sets without any prior smoothing process. Therefore, single-point regions can be identified.

All the alterations for 6 of the remaining cell lines covered a whole chromosome and would not be identified by analyzing each chromosome separately. However, analysis of an entire genome may help to identify whole chromosome alterations. We applied ODM to each of these cell lines with \( d = 4 \) standard deviation followed by one specific pruning step: If the absolute mean difference between 2 adjacent segments around a change point detected by ODM was less than 0.35, then that change point was not considered a detected change point. The results are shown in Figure 6 and clearly reveal whole chromosome alterations along each cell line.

6. Discussion and Conclusions

We have proposed several new approaches for assessing DNA copy number variation along a chromosome. These methods can be used to detect copy number alterations in tumors or other tissues. All approaches can be directly applied to analyze data sets without any prior smoothing process, and single-point
Table 3. Results from applying ODM, SDM, BRM, and CBS to 9 cell lines with known copy number alterations

<table>
<thead>
<tr>
<th>Cell line/Chromosomes</th>
<th>ODM 3.5σ</th>
<th>ODM 4σ</th>
<th>SDM α = 0.001 0.0005</th>
<th>BRM α = 0.001 0.0005</th>
<th>CBS α = 0.01 0.005</th>
</tr>
</thead>
<tbody>
<tr>
<td>GM03563/3</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>GM03563/9</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>GM03563/false(single)</td>
<td>8(2)</td>
<td>4(0)</td>
<td>7(2)</td>
<td>5(3)</td>
<td>6(3)</td>
</tr>
<tr>
<td>GM05296/10</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>GM05296/11</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>GM05296/false(single)</td>
<td>3(2)</td>
<td>1(1)</td>
<td>4(3)</td>
<td>3(3)</td>
<td>2(4)</td>
</tr>
<tr>
<td>GM01750/9</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>GM01750/14</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>GM01750/false(single)</td>
<td>3(1)</td>
<td>0(0)</td>
<td>3(1)</td>
<td>3(1)</td>
<td>1</td>
</tr>
<tr>
<td>GM03134/8</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>GM03134/false(single)</td>
<td>2(1)</td>
<td>0(1)</td>
<td>4(1)</td>
<td>1(1)</td>
<td>3(2)</td>
</tr>
<tr>
<td>GM13330/0</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>GM13330/4</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>GM13330/false(single)</td>
<td>6(0)</td>
<td>3(0)</td>
<td>8(1)</td>
<td>7(1)</td>
<td>8(1)</td>
</tr>
<tr>
<td>GM01535/5</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>GM01535/12</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>GM01535/false(single)</td>
<td>3(3)</td>
<td>1(1)</td>
<td>1(6)</td>
<td>1(6)</td>
<td>2(6)</td>
</tr>
<tr>
<td>GM07081/7</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>GM07081/15</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>GM07081/false(single)</td>
<td>3(3)</td>
<td>1(2)</td>
<td>4(4)</td>
<td>1(5)</td>
<td>2(5)</td>
</tr>
<tr>
<td>GM13031/17</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>GM13031/false(single)</td>
<td>5(0)</td>
<td>1(1)</td>
<td>7(1)</td>
<td>5(1)</td>
<td>5(2)</td>
</tr>
<tr>
<td>GM01524/6</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>GM01524/false(single)</td>
<td>6(1)</td>
<td>2(2)</td>
<td>5(1)</td>
<td>4(2)</td>
<td>4(2)</td>
</tr>
</tbody>
</table>

regions (or outliers) can be identified. We applied these methods to an array CGH data set of fibroblasts with known copy number changes confirmed by spectral karyotyping and were able to identify all the expected alterations. We also applied the methods to a 100-K SNP data set of non-small cell lung carcinoma, namely cell line H2122 from Zhao and others (2005). Our results were similar to Zhao and others (2005) for large regions, but we identified many more small regions or single-point alterations. The analysis time for the SNP data of 125 906 markers was very fast (less than 3 min on a 3.20-GHz PC in performing ODM). These methods are also likely to be useful in assessing normal variation. In addition, through simulation we have demonstrated that our methods can correctly detect change points that partition chromosomes into regions with the same underlying copy number.

In developing the methods, we have made the implicit assumption that residual variances (defined in (3.1)) are equal across the chromosome. In some additional simulations, we found that SDM still works well for heteroscedastic data as long as heteroscedasticity across different regions is mild. In general, if heteroscedasticity is apparent, we suggest that some pre-analysis should be performed to derive good estimates of variances. Then the signed distance measure in (2.1) could be modified to involve estimated variances, although this approach would require future investigation.

An issue that can arise with SDM is the edge effect caused by artificially connecting the first and the last points $Y_i$ and $Y_j$ in the segment $\{Y_i, \ldots, Y_j\}$. That is, if the index $k$ of an identified change point $Y_k$
Fig. 6. The ODM analysis of 6 whole fibroblast cell lines. Markers are arranged by chromosome. The vertical dotted lines demarcate chromosomes. The horizontal lines represent the mean of each segment.

is “close” to either $i$ or $j$, then it might not be a real change point. The same problem occurs for the CBS method (Olshen and others, 2004) and can be handled in a similar way. By contrast, in the process of change-point detection, BRM does not cut the whole data set into several segments so that this problem of small segments does not exist. Also, the edge effect is a minor issue for BRM because there is only one edge to be considered.

Any statistical method for analyzing DNA copy number will detect more change points or alterations than there are real copy number changes. These false positives may be caused by local trends in the data or other reasons that are not totally understood. Therefore, a pruning procedure may be helpful to eliminate some change points or to merge adjacent segments in order to reduce the number of false positives. Pruning can be performed either from a statistical viewpoint, by controlling a sequence of sum of squares as in CBS, or from a biological viewpoint by merging 2 adjacent segments as in the HMM method (Fridlyand and others, 2004) and the penalized least-squares regression method (Huang and others, 2005). Our methods do not include this step because any pruning procedure is dominated by a threshold parameter which must be determined by the type of data produced, the level of purity of the populations, and whether type I or type II error is more undesirable. As Fridlyand and others (2004) point out, threshold values for comparing the means of 2 adjacent segments could be as low as 0.25 when analyzing primary tumors and as high as 0.5 when looking for genetic abnormalities homogeneously present in all cells. For the results given in Table 3, CBS includes a pruning procedure by controlling a sequence of sum of squares, while our 3 methods do not perform any pruning.
It may be noted that the critical values $\alpha = 0.001, 0.0005$ used for SDM and BRM in both simulated and real data studies are much smaller than the ones used in CBS ($\alpha = 0.01, 0.005$). This is because all our change-point detection methods are essentially based on identifying the outliers among the scores produced by CCTTS algorithm, and this is conceptually different from the CBS approach. CBS, using the likelihood ratio, identifies only the largest change point at each segmentation step, whereas our methods produced by CCTTS algorithm, and this is conceptually different from the CBS approach. CBS, using the all our change-point detection methods are essentially based on identifying the outliers among the scores and real data studies are much smaller than the ones used in CBS ($\alpha$ for each permutation. That means that the distribution $Fs$ can be used to further identify whole chromosome gain or loss.

Inferences about change points drawn from analyzing the whole genome, rather than single chromosomes, methods can also analyze an entire genome simultaneously to identify regions of copy number variation. To analyze each chromosome separately due to the fact that chromosomes are physically separated. Our CBS. Furthermore, more markers means that even fewer permutations are needed. Existing methods tend give comparable performance, but the interpretation is not the same.

We have shown that, for normally distributed data, the ODM is easy to perform and extremely fast. The numbers of permutations needed for both SDM and BRM are small compared with the number needed for CBS. Furthermore, more markers means that even fewer permutations are needed. Existing methods tend to analyze each chromosome separately due to the fact that chromosomes are physically separated. Our methods can also analyze an entire genome simultaneously to identify regions of copy number variation. Inferences about change points drawn from analyzing the whole genome, rather than single chromosomes, can be used to further identify whole chromosome gain or loss.

ACKNOWLEDGMENTS

This work was supported by the Mathematics of Information Technology and Complex Systems (MITACS/NCE), by National Institutes of Health grant HL68532, and Genome Canada through the Center for Applied Genomics, Toronto. Conflict of Interest: None declared.

APPENDIX A

A.1 Simple example illustrating the clustering algorithm and property (a)

We consider a simple situation of a data set containing only 3 points in sequence: $Y_1, Y_2, and Y_3$. Let $D_1 = D((Y_1), \{Y_2\}), D_2 = D((Y_2), \{Y_3\}), and D_3 = D((Y_3), \{Y_1\})$, where $D$ is defined by (2.1). At the beginning of the CCTTS algorithm, the initial scores are $S_1 = D_1, S_2 = D_2$, and $S_3 = D_3$. We focus here on how to obtain the final score of $S_1$. According to the CCTTS algorithm, we have

1) When $|D_1| = |D_2| \leq |D_3|$ or $|D_1| = |D_3| \leq |D_2|$ or $|D_2| = |D_3| \leq |D_1|$, then more than 2 clusters are merged simultaneously. One big cluster is obtained, and $S_1$ is not updated. So the final score is

$$S_1 = D_1.$$  

2) When $|D_1| < \min(|D_2|, |D_3|)$, then we merge clusters $\{Y_1\}$ and $\{Y_2\}$, and we update $S_2$ and $S_3$, but not $S_1$. Since $\{Y_1\}$ and $\{Y_2\}$ are merged together, $S_1$ will not be updated anymore. Therefore, the final score is

$$S_1 = D_1.$$  

3) When $|D_2| < \min(|D_1|, |D_3|)$, then we merge clusters $\{Y_2\}$ and $\{Y_3\}$, and we update $S_1$ to obtain

$$S_1 = \begin{cases} D_1, & |D_1| > D((Y_1), \{Y_2, Y_3\}), \\ D((Y_1), \{Y_2, Y_3\}), & |D_1| \leq D((Y_1), \{Y_2, Y_3\}). \end{cases}$$

A hierarchical clustering method for estimating copy number variation 649

Conflict of Interest: None declared.
Note that, at this stage, the data set contains 2 clusters. The next merging procedure will produce one big cluster, and hence statistic scores will not be updated anymore. Therefore, the above achieved value is the final score of $S_1$.

4) When $|D_3| < \min(|D_1|, |D_2|)$, then we merge clusters $\{Y_3\}$ and $\{Y_1\}$. In a similar discussion as in (3), the final score of $S_1$ can be obtained by

$$S_1 = \begin{cases} D_1, & |D_1| > |D((Y_3, Y_1), \{Y_2\})|, \\ D((Y_3, Y_1), \{Y_2\}), & |D_1| \leq |D((Y_3, Y_1), \{Y_2\})|. \end{cases}$$

By summarizing (1), (2), (3), and (4), $S_1$ can be mathematically expressed as

$$S_1 = \begin{cases} D_1 = D((Y_1), \{Y_2\}), & \text{if } |D_1| = |D_2| \leq |D_3|, \text{ or } |D_1| = |D_3| \leq |D_2|, \\ & \text{or } |D_2| = |D_3| \leq |D_1|, \text{ or } |D_1| < \min(|D_2|, |D_3|), \\ & \text{or } |D_2| < \min(|D_1|, |D_3|) \text{ and } |D_1| > |D((Y_1), \{Y_2, Y_3\})|, \\ & \text{or } |D_3| < \min(|D_1|, |D_2|) \text{ and } |D_1| > |D((Y_3, Y_1), \{Y_2\})|, \\ & |D_3| < \min(|D_1|, |D_2|) \text{ and } |D_1| \leq |D((Y_3, Y_1), \{Y_2\})|, \\ & |D_3| < \min(|D_1|, |D_2|) \text{ and } |D_1| \leq |D((Y_3, Y_1), \{Y_2\})|. \end{cases}$$

Similarly, we can obtain mathematical expressions for $S_2$ and $S_3$. For example, $S_2$ can be written as

$$S_2 = \begin{cases} D_2 = D((Y_2), \{Y_3\}), & \text{if } |D_2| = |D_3| \leq |D_1|, \text{ or } |D_2| = |D_1| \leq |D_3|, \\ & \text{or } |D_3| = |D_1| \leq |D_2|, \text{ or } |D_2| < \min(|D_3|, |D_1|), \\ & \text{or } |D_3| < \min(|D_2|, |D_1|) \text{ and } |D_2| > |D((Y_2), \{Y_3, Y_1\})|, \\ & \text{or } |D_1| < \min(|D_2|, |D_3|) \text{ and } |D_2| > |D((Y_1, Y_2), \{Y_3\})|, \\ & |D_3| < \min(|D_2|, |D_1|) \text{ and } |D_2| \leq |D((Y_2), \{Y_3, Y_1\})|, \\ & |D_3| < \min(|D_2|, |D_1|) \text{ and } |D_2| \leq |D((Y_1, Y_2), \{Y_3\})|. \end{cases}$$

From this simple case, we find that (i) the final statistic scores $S_i$ from the CCTTS algorithm are functions of the original sequence data set and (ii) if $S_1$ is written as $S_1 = f(Y_1, Y_2, Y_3)$, then $S_2 = f(Y_2, Y_3, Y_1)$. So, $S_1$ and $S_2$ are identically distributed when $Y_1, Y_2,$ and $Y_3$ are iid.

**APPENDIX B**

B.1 *Conditions for asymptotically unbiased variance estimation*

Let $S_1, \ldots, S_N$ be the statistic scores computed from the CCTTS. According to property (b) of $S_i$ being approximately identically normally distributed, negative correlations between 2 adjacent statistics, and approximate independence between statistics at least 2 positions apart, we simplify the assumptions for the joint distribution of $S_1, \ldots, S_N$ as follows:

1) Marginal distributions:

$$S_i \sim N(\mu, \sigma^2), \quad i = 1, \ldots, N.$$
2) Correlations:

\[
\text{Corr}(S_i, S_{i+j}) = \begin{cases} 
\rho, & i = 1, \ldots, N - 1, j = 1, \\
or i = 1, j = N - 1, \\
0, & i = 1, 1 < j < N - 1, \\
or i = 2, \ldots, N - 2, 1 < j \leq N - i.
\end{cases}
\]

Here, the assumption of the correlation \(\text{Corr}(S_1, S_N) = \rho\) is based on the circular structure.

The sample variance \(\hat{\sigma}^2\) from \(S_1, \ldots, S_N\) is given by

\[
\hat{\sigma}^2 = \frac{1}{N-1} \sum_{i=1}^{N} (S_i - \bar{S})^2,
\]

where \(\bar{S} = \frac{1}{N} \sum_{i=1}^{N} S_i\). Our goal is to prove that \(\hat{\sigma}^2\) is an asymptotically unbiased estimator of \(\sigma^2\) with respect to \(N\) under the assumptions given in (1) and (2). This can be done by the following derivation:

\[
E(\bar{S}^2) = \frac{1}{N^2} \left\{ \sum_{i=1}^{N} E(S_i^2) + 2 \sum_{i < j} E(S_i S_j) \right\} = \frac{1}{N^2} \left\{ N(\sigma^2 + \mu^2) + 2 \left[ \sum_{i=1}^{N-1} E(S_i S_{i+1}) + E(S_1 S_N) \right. \right.
\]
\[
\left. + \sum_{j=3}^{N-1} E(S_1 S_j) + \sum_{i=2}^{N-2} \sum_{j=i+2}^{N} E(S_i S_j) \right\} = \frac{1}{N^2} \left\{ N(\sigma^2 + \mu^2) + 2N(\mu^2 + \rho \sigma^2) + (N^2 - 3N)\mu^2 \right\} = \frac{1}{N} \{ \sigma^2 + N\mu^2 + 2\rho \sigma^2 \}.
\]

Therefore,

\[
E(\hat{\sigma}^2) = \frac{1}{N-1} \left[ \sum_{i=1}^{N} E(S_i^2) - N E(\bar{S}^2) \right] = \frac{1}{N-1} \left[ N(\sigma^2 + \mu^2) - (\sigma^2 + N\mu^2 + 2\rho \sigma^2) \right] = \sigma^2 - \frac{2\rho \sigma^2}{N-1} = \sigma^2 \left( 1 - \frac{2\rho}{N-1} \right).
\]

\(E(\hat{\sigma}^2) > \sigma^2\) when \(\rho < 0\) (negative correlation), and \(E(\hat{\sigma}^2) \to \sigma^2\) as \(N \to \infty\).
REFERENCES


[Received March 20, 2006; revised September 1, 2006; accepted for publication October 16, 2006]